Processing time tolerance-based ACO algorithm for solving job-shop scheduling problem

Yabo Luo*, Yongo P Waden

School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan city, 430070, China

*Corresponding author's email: luoyabo1973@163.com

Abstract. Ordinarily, Job Shop Scheduling Problem (JSSP) is known as NP-hard problem which has uncertainty and complexity that cannot be handled by a linear method. Thus, currently studies on JSSP are concentrated mainly on applying different methods of improving the heuristics for optimizing the JSSP. However, there still exist many problems for efficient optimization in the JSSP, namely, low efficiency and poor reliability, which can easily trap the optimization process of JSSP into local optima. Therefore, to solve this problem, a study on Ant Colony Optimization (ACO) algorithm combined with constraint handling tactics is carried out in this paper. Further, the problem is subdivided into three parts: (1) Analysis of processing time tolerance-based constraint features in the JSSP which is performed by the constraint satisfying model; (2) Satisfying the constraints by considering the consistency technology and the constraint spreading algorithm in order to improve the performance of ACO algorithm. Hence, the JSSP model based on the improved ACO algorithm is constructed; (3) The effectiveness of the proposed method based on reliability and efficiency is shown through comparative experiments which are performed on benchmark problems. Consequently, the results obtained by the proposed method are better, and the applied technique can be used in optimizing JSSP.

1 Introduction

Generally, the job shop scheduling problem (JSSP) is known as NP-hard problem, which is characterized by uncertainty and complexity that cannot be tackled by a linear method[1]. In the mid-60s to the early 70s, scheduling problem was studied by many researchers. Thereupon, scheduling problem was proved to be a NP-hard problem, which is unsolvable in the polynomial time, although burgeoning heuristic algorithms were utilized to solve scheduling problem. Afterward, the invention of bionic algorithms and their application in scheduling problem have resulted with a success. For instance, Tabu search[2], simulated annealing[3], genetic algorithm[4], particle swarm optimization[5], and ACO algorithm, have been widely used to tackle JSSP.

In spite of successful research results on the JSSP, challenges and drawbacks still existed on the JSSP, for example:

(1) Traditional optimization methods and heuristic algorithms have their own shortcomings and limitations. Moreover, the methods cannot solve large-scale problems, whereas, the new methods have some disadvantages such as poor convergence and low search efficiency.

(2) Generally, the bionic algorithm still lacks rigorous theoretical verification, mainly because they can be easily trapped in a local optimum with their instability in convergence.

To solve the aforementioned problems, this study is carried out on improving the ACO algorithm by introducing a constraint satisfying tactic in order to handle JSSP efficiently. Hence, the proposed method will advance JSSP in a practical sense.

2 Constraint analysis of JSSP

The feature model of general JSSP is described as follows: there are N workpieces needing to be processed on M machine. The process sequences of the workpiece on the machine, processing time of each process, and the time of a workpiece on each machine are determined. Therefore, the objective of scheduling is to arrange the process sequences on the machines and decide on the start time of each process in order to find an optimal objective function which is known as makespan.

Commonly, JSSP is described by job processing on the matrix D and processing time on the matrix T. For example, the constraints for a JSSP, made of three workpiece and three machines, can be as follows:

(1) Machine capacity constraint.

A piece of machine can only process one process of a workpiece at one time, and when arranging the process time to a machine, the processing time cannot have overlapping points.

(2) Task delivery constraint.

The completion time of each workpiece should be less than or equal to the due time.

(3) Process flow constraint.

Each task is made up of a series of processes, between which exist a series or a parallel relationship according to the requirements of process flow.

Consequently, the constraints in JSSP must be satisfied to guarantee the feasibilities of scheduling results. Firstly, constraints should be satisfied with the process of searching a solution, whereas, feasible solution of JSSP is also a feasible scheme, which includes the processing arrangements of machines and the starting time of each process. Finally, the solutions of a JSSP can be represented by a set.

3 Improved ACO algorithm based on constraint satisfied tactics

The constraint satisfied technology is used to improve the search speed, where the variables to be improved in JSSP are the constraints, which are made of the process sequence constraints and the equipment resource constraints. Nevertheless, the constraint satisfied technology mainly focuses on the consistency processing technology and the constraint spread algorithm.

(1) Consistency Processing Technology (CPT).

Mainly, Consistency Processing Technology (CPT) is a kind of control technologies in advance, which processes the constraints of Constraint Satisfaction Problem (CSP) in the beginning of solving a problem. As a consequence, a search space is pruned according to the initial conditions or the last search results by removing the values that don’t satisfy the constraints in current search space.

Equation (1) is for pheromone updating, among which, ρ is the lasting coefficient of pheromone; $\Delta\tau_{ij}$ is the quantity of pheromone that kth ant left at path (i, j) in this iteration; Q is a positive constant; L_k is the path length kth ant covers in this search.

$$\begin{aligned}
&\tau_{ij}(t+n) = \rho * \tau_{ij}(t) + \Delta\tau_{ij} \\
&\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k} \\
&\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k}, \text{ if ant k passes by edge ij} \\ 0, \qquad\qquad \text{else} \end{cases}
\end{aligned} \tag{1}$$

Therefore, after the consistency processing, the search space will continue to shrink, and consequently the subsequent search will also reduce many steps. Furthermore, the search efficiency is improved correspondingly. For example, a JSSP system with 3 workpieces, each have three serial processes, 9 processes of 3 workpieces all enter the search space in the first step of search. Afterward, only 3 initial processes of the consistency processing will remain, whereas, the others which do not meet the process sequence constraints are removed.

(2) Constraint Spread Algorithm (CSA).

In the Constraint Spread Algorithm (CSA) the search space is reduced to enhance the consistency of constraints during the search process. When solving a problem of the arrangement process of JSSP, the CSA is mainly applied to realize the sequence constraint spread of processes and the constraint spread of resources.

The process of workpiece changes forward dynamically along with the process route. After the current process is completed, the next one can enter the current processing sequence. Thus, the spread of processes must follow the requirements of process routes. Therefore, the constraint spread of process sequence is mainly to judge the operation time and the starting time of the former process, and disperse down to find the primary time of the next one.

Setting $\tau_{ij}(0) = C$, C is a small constant. The pheromone of each path will be updated according to equation (2) when ants complete a search.

$$\tau_{ij}(t+n) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}$$
$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k} \qquad (2)$$

ρ is the lasting coefficient of pheromone on the path. 1-ρ is the evaporation coefficient of pheromone;

$\Delta\tau_{ij}$ is the increment of pheromone on path ij in this iteration;

$\Delta\tau_{ij}^{k}$ is the quantity of pheromone left on path ij by the kth ant in this iteration.

The constraints of resource are mainly performed on the equipment. For instance, when a different process of workpiece requires a certain piece of equipment, the constraint satisfied problem on equipment is induced. Normally, the Constraint Spread of Resources (CSR) is devised with the time sequences of available equipments. In an optional process, the CSR is used to judge the equipment. If the constraint is satisfied, the equipment will enter processing. Otherwise, it will be excluded from instance search space, which will wait for the next time.

(3) The steps of improved ACO algorithm.

An improved ACO algorithm mainly exploits constraint satisfaction technology to prune the search space in order to enhance the search efficiency of ACO algorithm. Hence, the algorithm is improved by exploiting global and local pheromone updating strategy. Additionally, the situation is revised in which the algorithm can easily stagnate. Specific steps are as follows:

Step 1: Initialization.

The scheduling information is made of the processing time, required equipment, and algorithm parameters, including the number of ants, iteration times and the coefficient of pheromone.

Step 2: Consistency processing.

The objective is to judge process constraints and equipment constraints which have entered scheduling sequence, and remove conflicting tasks. Whereas, the current tasks are process sets for meeting the process constraints and the equipment constraints.

Step 3: Choose the next node according to state transition rule.

Here, the tasks obtained from consistency processing can be scheduled immediately, and then the next scheduling, task can be selected according to state transition rule in the ant colony algorithm.

Step 4: Constraint spread algorithm.

Process and available equipment are allowed to change accordingly. Nevertheless, in the non-scheduling process and available equipment, in any conflict, the judge constraint will take control on the equipment and on the process. If there is no conflict, it executes step 5, and returns to the step 2.

Step 5: Judge whether all tasks have been scheduled.

If all tasks have been scheduled, it turns to step 6; if there is a task not scheduled, it turns to step 2, and continues to carry out scheduling process.

Step 6: Judge whether the search is over.

The iteration is raised after the updating of pheromone. If the time of iteration reaches the maximum, the algorithm ends, and prints results.

In this paper, minimum makespan is considered the objective function, and the input parameters are mainly divided into two parts, namely, information processing: number of tasks, process route, available equipment, and the processing time. Parameters of the algorithm: number of ants, pheromone evaporation coefficient, heuristic factor, pheromone factor, iteration times and the total amount of pheromone.

4 Comparative experimental study

In this section, we have performed a comparative experiment by adopting well known benchmark problems. These problems are good for testing the performance of new methods, and therefore, feasibility of the proposed method is tested on the following: a scheduling problem which has 9 tasks, 3 workpiece, 3 processes for every workpiece, and 5 available machines. The specific processing time and the corresponding equipment are as shown in Table 1.

Table 1: Required processing times and equipment information for tasks.

Tasks Number	101	102	103
Required Time	5	8	10
Required Equipment	A	B	C
Equipment Number	A-2	B-2	C-1
Tasks Number	201	202	203
Required Time	10	6	5
Required Equipment	B	A	C
Equipment Number	1	2	3
Tasks Number	301	302	303
Required Time	5	10	7
Required Equipment	C	B	A
Equipment Number	4	5	6

Figure 1 shows the disjunction graph for the process sequence constraints between each node in which the nodes are linked by the black lines in the disjunctive graph. Furthermore, ants cannot move against the expect direction of the black solid line when they are at a certain node and moving to the next node. However, when in the large problems, the ants' choice will lead to the increase of search time and the decrease of efficiency. Nonetheless, as the problem expands, the complexity of calculation grows exponentially. Additionally, a solution to this problem can mainly depend on implementing the constraint satisfied technology, through consistency processing and constraint propagation. Thus, pruning the constraint space is dynamic, whereas, search time is stored in order to improve search efficiency.

In the disjunctive graph in Figure 1, the dotted lines link the tasks. Using the same equipment, the dotted lines also confirm priority selection of the equipment for a given task. Furthermore, from the state transition rule in the ACO algorithm, the ant with a larger probability occupies the equipment. Figure 3 is a scheduling result of the Gantt chart.

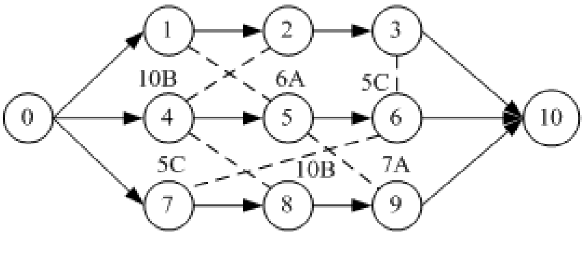

Figure 1: Disjunctive graph model

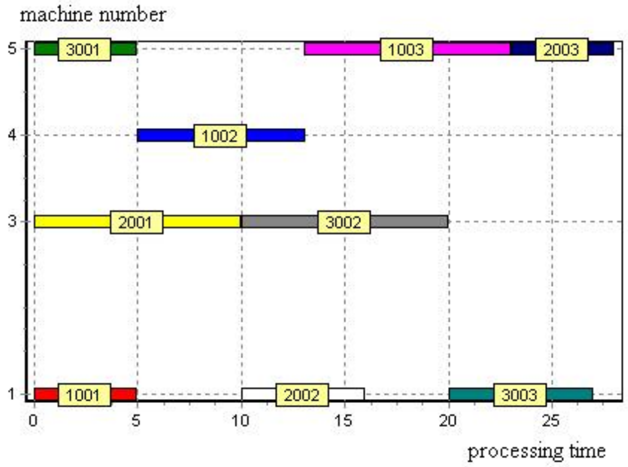

Figure 2: The Gantt chart of scheduling results

As it can be seen in the Figure 2, when the algorithm is at the beginning of the search, because of no guideline on the pheromone, the selected path may not be the shortest, in which it affects other ants' selection in a short term. However, other routes can be found during the search. If a shorter path is found, the pheromone on the path enhances constantly which can attract more ants to choose the same path. The running results indicated that the ACO algorithm can maintain its robustness and stability and converge to the optimal solution faster.

The result of the aforementioned benchmark problems is shown in Figure 3, which is obtained by using a traditional optimization algorithm. As it is seen in the graph, the traditional method was not able to obtain the best value in 50 iterations, whereas, the improved ACO was able to obtain the best value 28 in only 10 iterations. It can be concluded that the traditional optimization method cannot find satisfactory results for a small scale in the JSSP.

The iterative results have revealed that the best value can be obtained when iterations is increased to 700 iterations, where the near optimal solution can be achieved within 500-700 iterations. Furthermore, to prevent the trap of the algorithm in local optima, the update of pheromone is performed, which is carried out by combining the local and global. Hence, it can extend the search time, as well as the iterations.

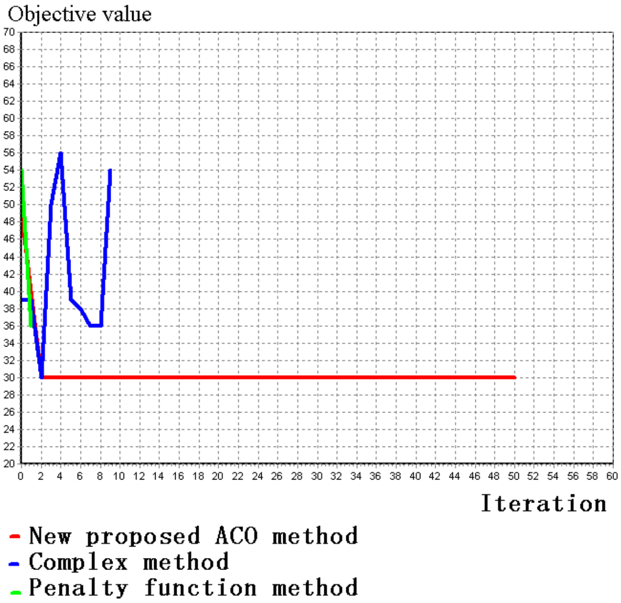

Figure 3: The comparative experimental results

The stagnation of the algorithm can be avoided as can be seen in Figure 3. Moreover, the results obtained by the improved algorithm are better than those of traditional methods, which is because of the combined effect of pheromone and heuristic factors. The algorithm can only obtain the sub optimal solution, however, it can be near the optimal solution. Therefore, no large deviation is observed and for that reason the algorithm is robust.

In this work, ACO is improved and its performance has shown competence over the traditional method as the results have indicated. Thus, the proposed method can be applied for solving scheduling in the JSSP.

5 Conclusions

In order to obtain a feasible solution, complexity is reduced by using an improved ACO to handle JSSP. To our knowledge, no any study has been carried out on the ACO algorithm based on CSP. Therefore, a study on the constraint satisfied tactics with an ACO algorithm to improve the efficiency and reliability of the ACO algorithm is conducted. Afterward, the concept is implemented on the ACO. The main contributions of the research are as follows:

(1) New constraint model based on CSP is developed for the JSSP, in which, the techniques of tactics are employed, and the feedback is used to improve the ACO algorithm. As a consequence a better reliability is achieved.

(2) Constraint satisfied tactics are used to modify the slow convergence speed of the ACO algorithm. Further, a search space is reduced by using consistency processing and constraint spread algorithm. A search space is narrowed by removing the nodes that do not match with the instance constraints. Additionally, the search efficiency is improved.

The comparative experiments have shown the effectiveness of the proposed method, where it is superior in reliability and efficiency.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (NSFC.51375357), Independent Innovation Research Fund (2016-zy-018) for graduate student in Wuhan University of Technology, and the China Scholarship Council (2014GF099).

References

[1] Luo Yabo. “Topological sorting-based two-stage nested ant colony algorithm for job-shop scheduling problem”, *Journal of Mechanical Engineering*, 51, pp.178-184, (2015).

[2] Rohaninejad Mohammad, Kheirkhah Amir Saman, Vahedi Nouri Behdin, Fattahi Parviz. “Two hybrid tabu search-firefly algorithms for the capacitated job shop scheduling problem with sequence-dependent setup cost”, *International Journal of Computer Integrated Manufacturing*, 28, pp.470-487, (2015).

[3] Shivasankaran N, Senthil Kumar P, Venkatesh Raja K. “Hybrid sorting immune simulated annealing algorithm for flexible job shop scheduling”, *International Journal of Computational Intelligence Systems*, 8, pp.455-466, (2015).

[4] Driss Imen, Mouss Kinza Nadia, Laggoun Assia. “A new genetic algorithm for flexible job-shop scheduling problems”, *Journal of Mechanical Science and Technology*, 29, pp.1273-1281, (2015).

[5] Zheng Yuyan, Qu Jianhua, Wang Lin. “An improved cooperative PSO algorithm for job-shop scheduling problem”, *Lecture Notes in Computer Science*, 8944, pp. 265-277, (2015).